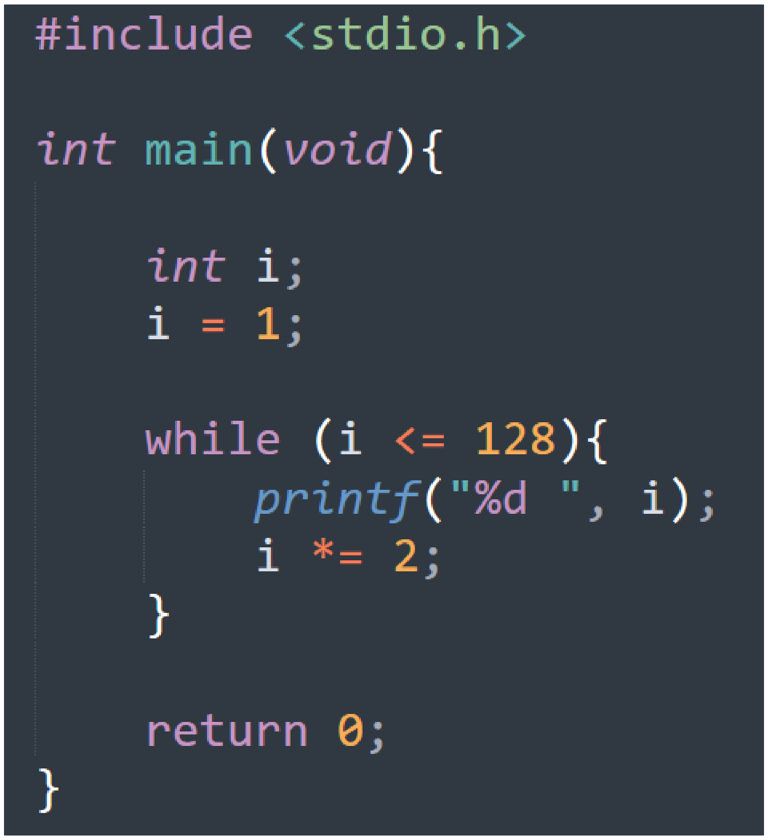
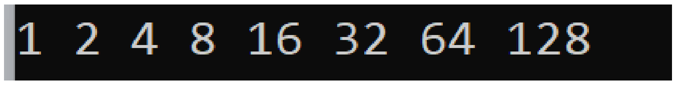
John Romson E. Erazo
2022-13004
BS in Computer Science I

1)



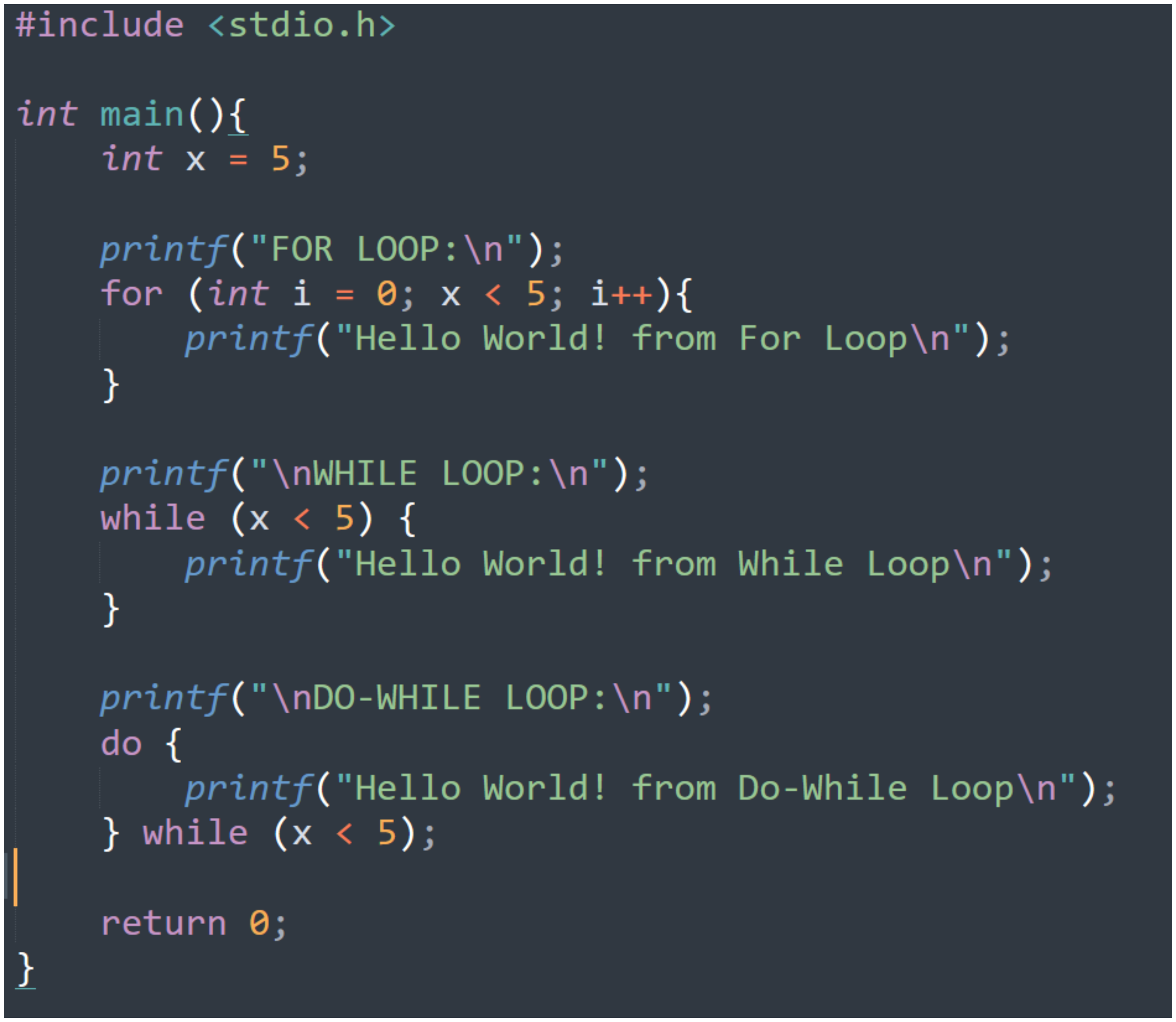
```c
#include <stdio.h>

int main(void){

    int i;
    i = 1;

    while (i <= 128){
        printf("%d ", i);
        i *= 2;
    }

    return 0;
}
```



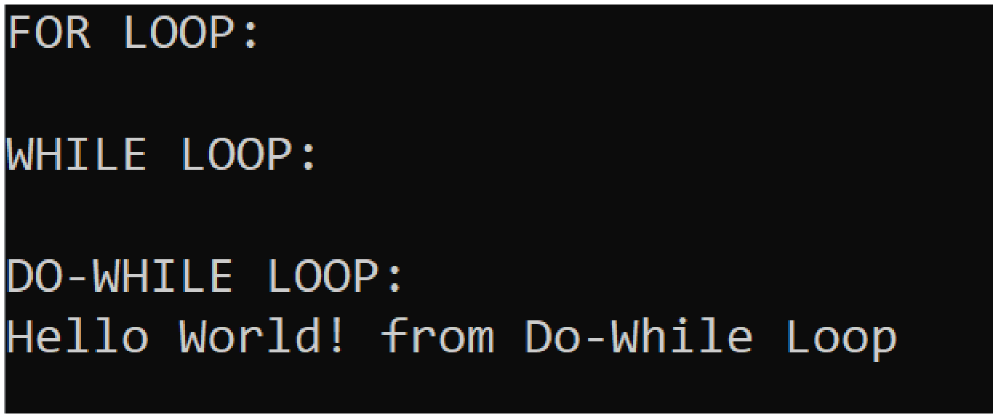
```
1 2 4 8 16 32 64 128
```

The output of the program is 1 2 4 8 16 32 64 128

2)



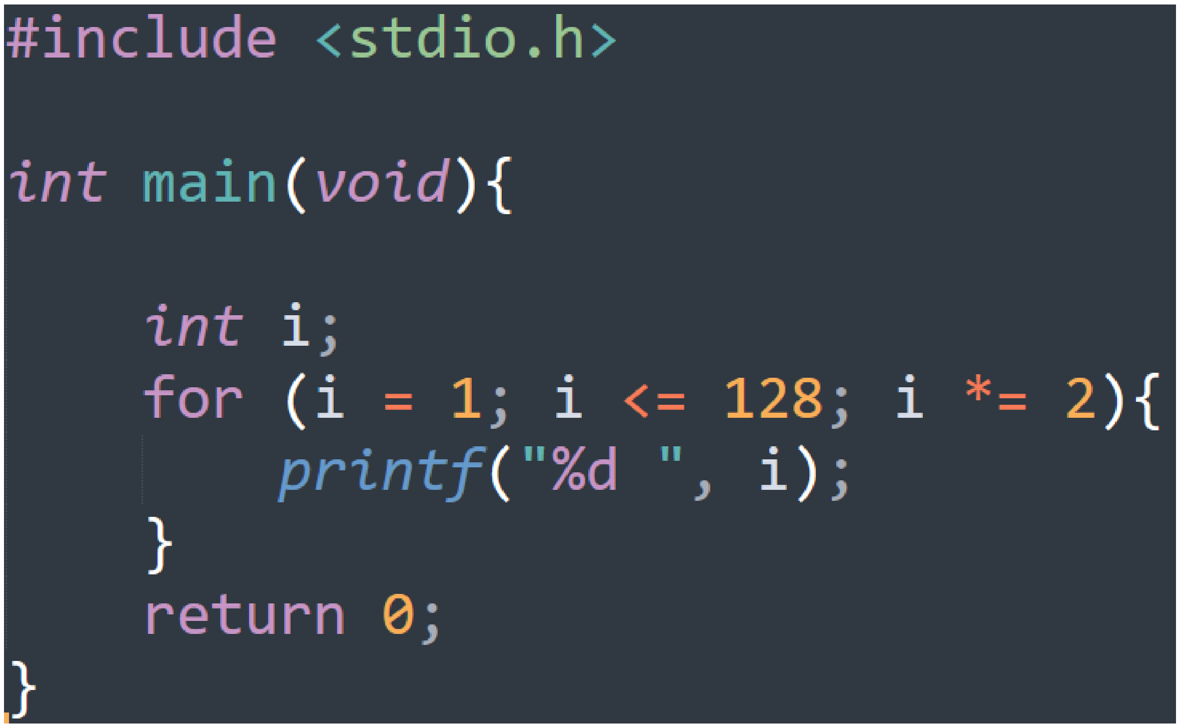
```c
#include <stdio.h>

int main(){
    int x = 5;

    printf("FOR LOOP:\n");
    for (int i = 0; x < 5; i++){
        printf("Hello World! from For Loop\n");
    }

    printf("\nWHILE LOOP:\n");
    while (x < 5) {
        printf("Hello World! from While Loop\n");
    }

    printf("\nDO-WHILE LOOP:\n");
    do {
        printf("Hello World! from Do-While Loop\n");
    } while (x < 5);

    return 0;
}
```



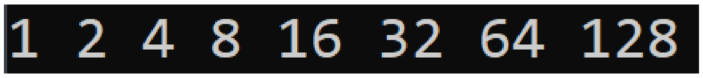
```
FOR LOOP:

WHILE LOOP:

DO-WHILE LOOP:
Hello World! from Do-While Loop
```

Letter c, the do-while loop is not equivalent to the for and while loop since the do-while loop guarantees that the loop body is executed at least once before checking the condition.

3)



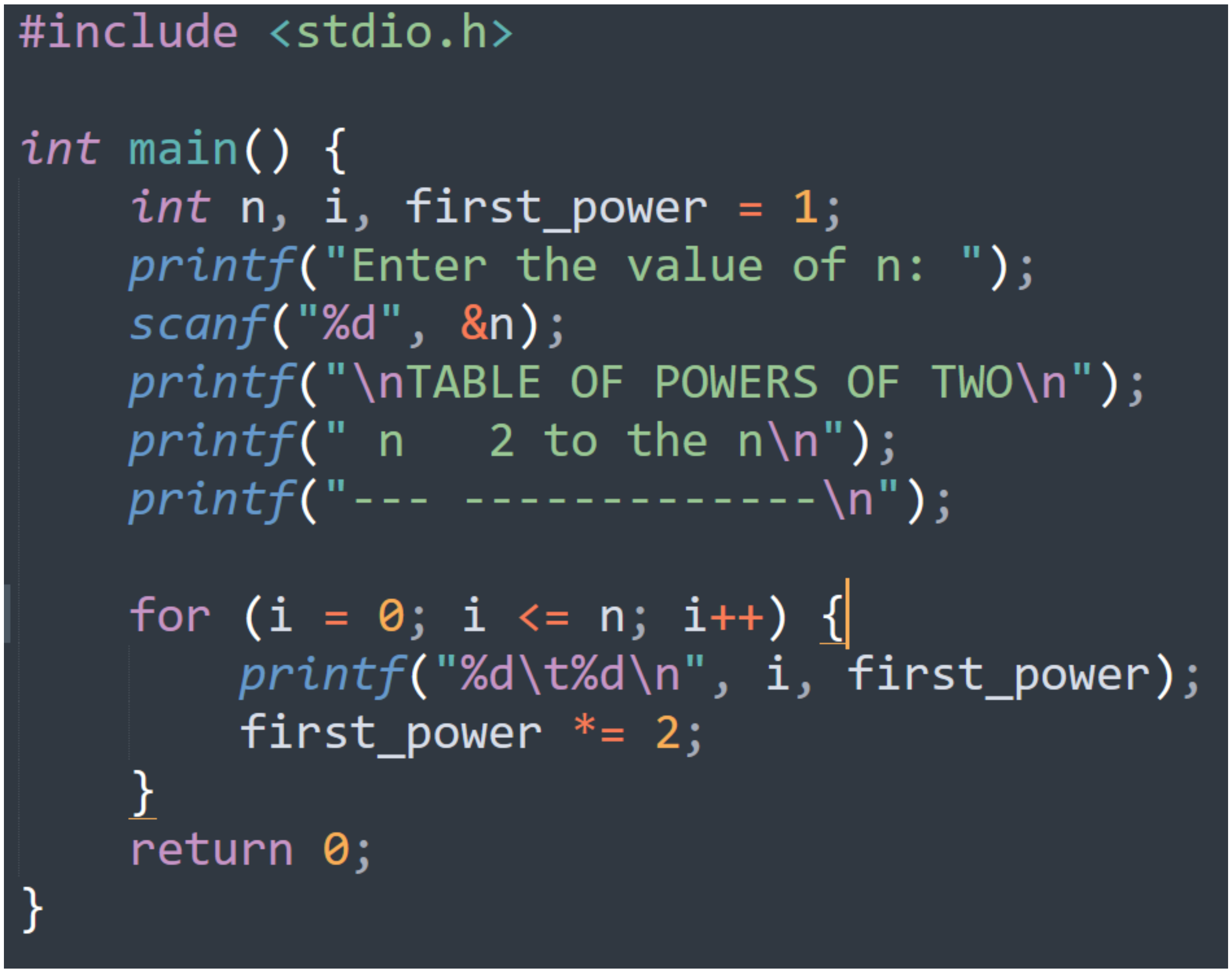
```c
#include <stdio.h>

int main(void){

    int i;
    for (i = 1; i <= 128; i *= 2){
        printf("%d ", i);
    }
    return 0;
}
```



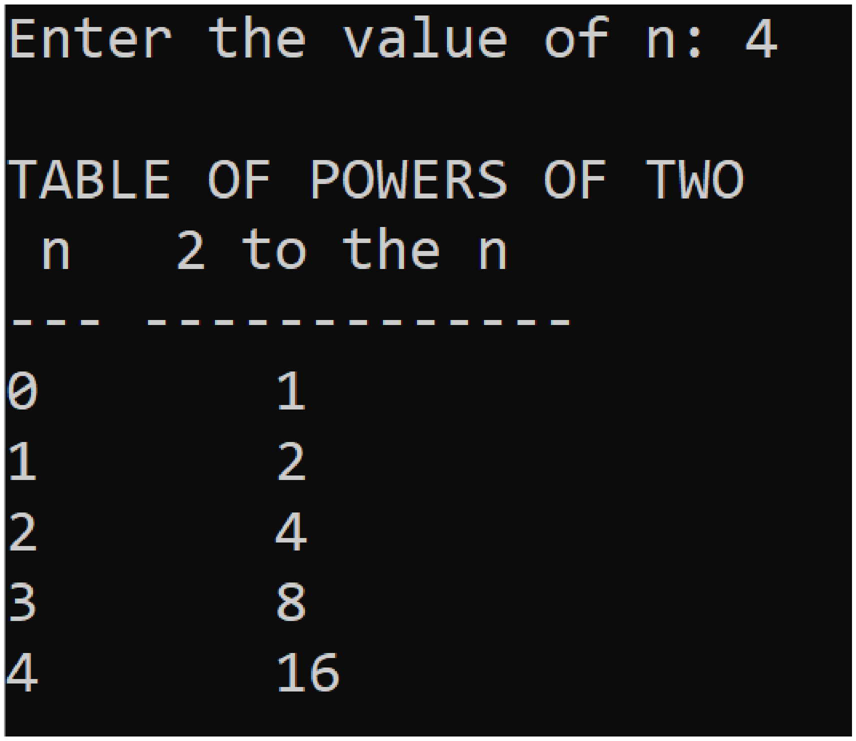
```
1 2 4 8 16 32 64 128
```

The output of the program is still the same: 1 2 4 8 16 32 64 128

4)

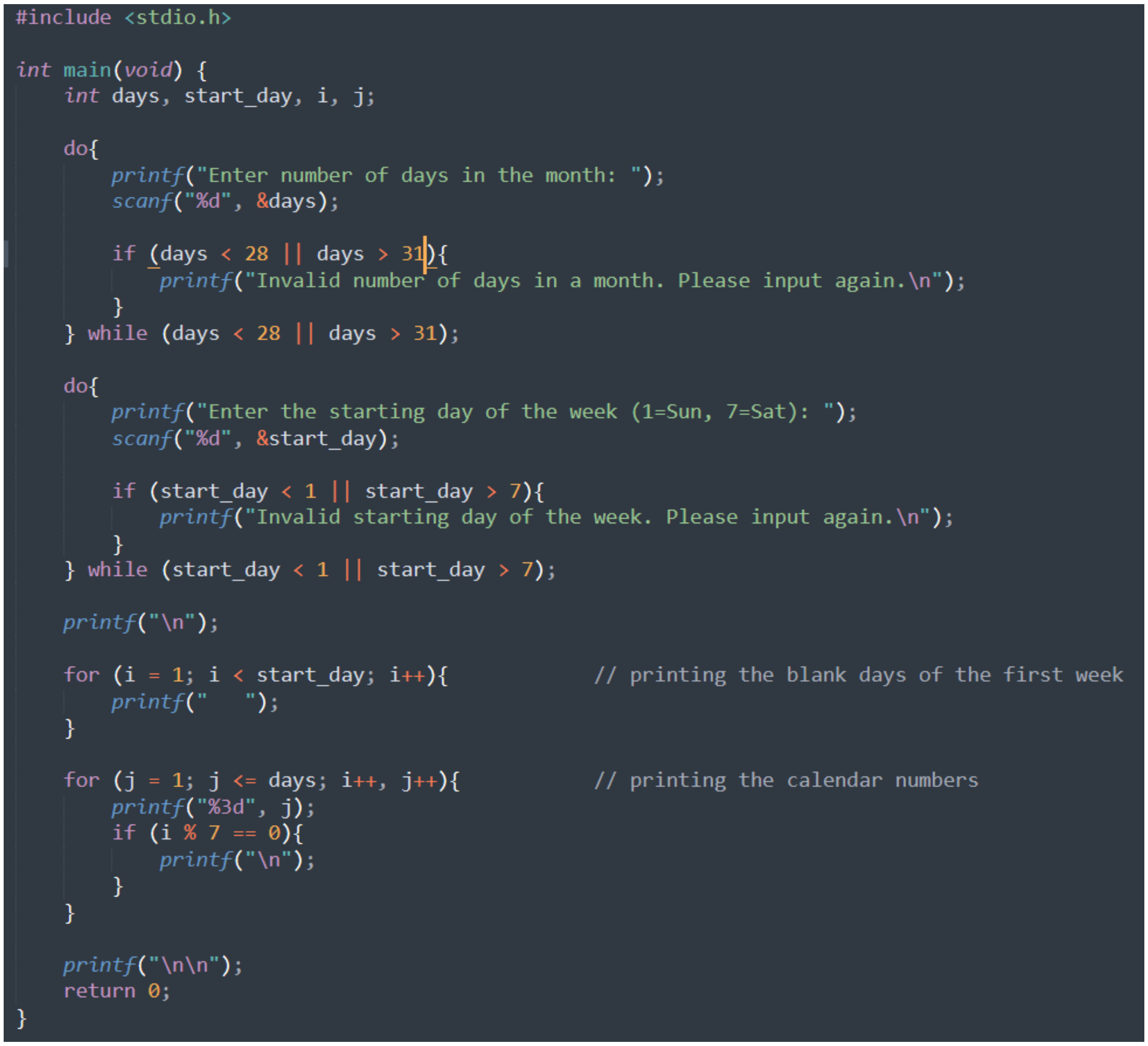


```c
#include <stdio.h>

int main() {
    int n, i, first_power = 1;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    printf("\nTABLE OF POWERS OF TWO\n");
    printf(" n   2 to the n\n");
    printf("--- -------------\n");

    for (i = 0; i <= n; i++) {
        printf("%d\t%d\n", i, first_power);
        first_power *= 2;
    }
    return 0;
}
```

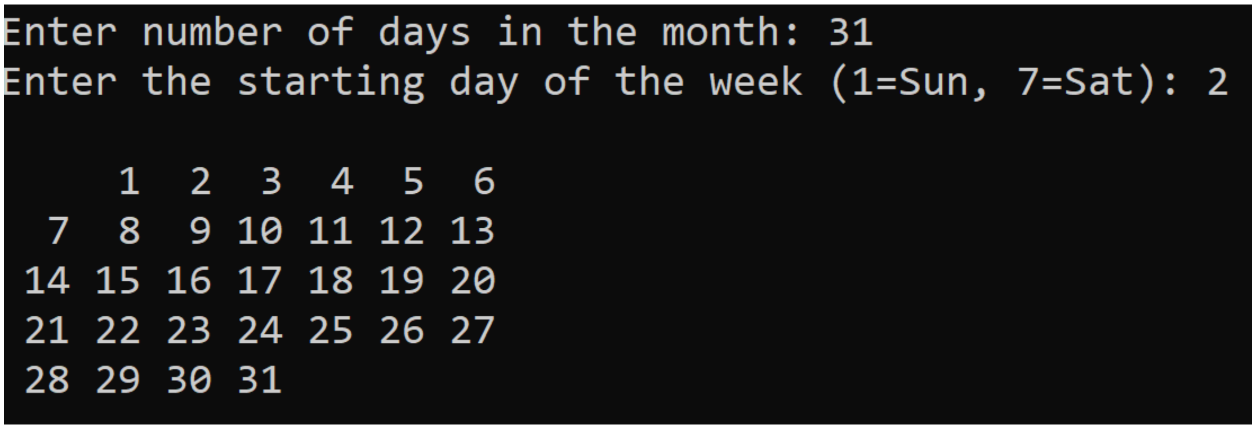


```
Enter the value of n: 4

TABLE OF POWERS OF TWO
 n   2 to the n
--- -------------
0       1
1       2
2       4
3       8
4       16
```

5)

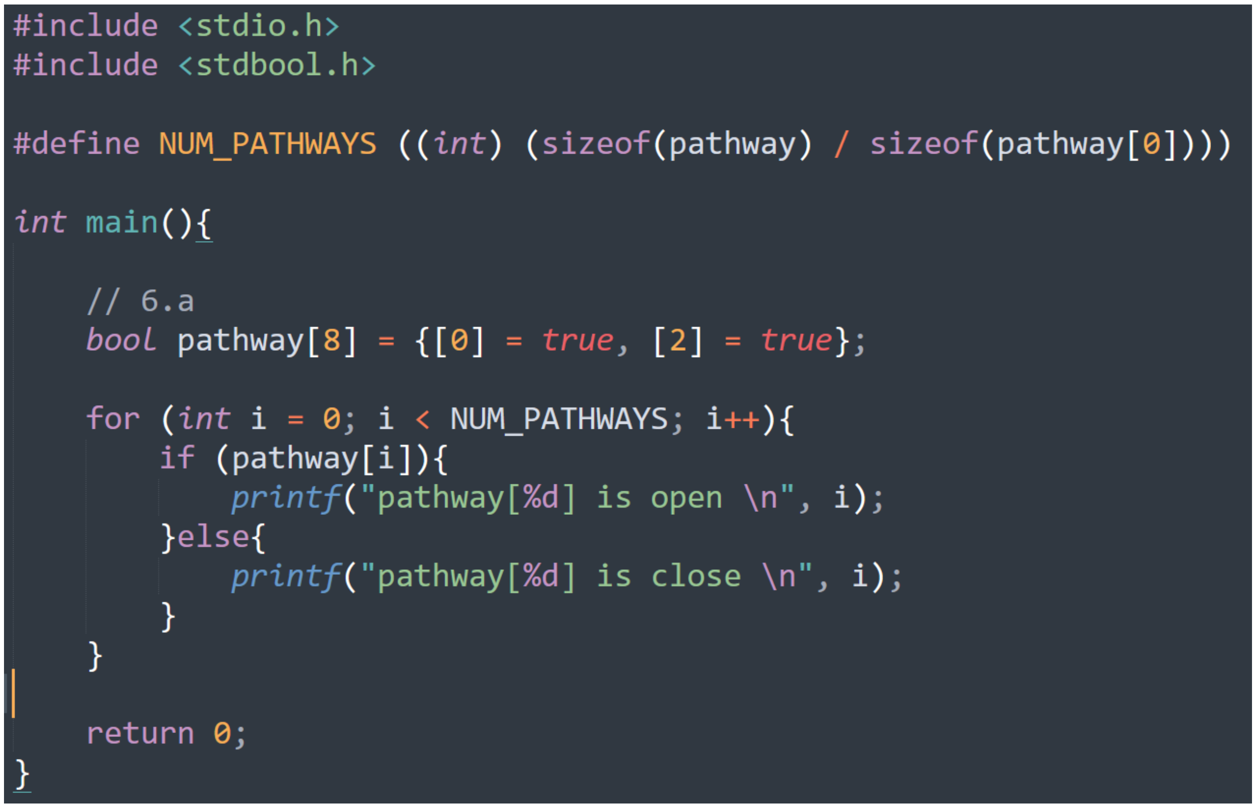


```c
#include <stdio.h>

int main(void) {
    int days, start_day, i, j;

    do{
        printf("Enter number of days in the month: ");
        scanf("%d", &days);

        if (days < 28 || days > 31){
            printf("Invalid number of days in a month. Please input again.\n");
        }
    } while (days < 28 || days > 31);

    do{
        printf("Enter the starting day of the week (1=Sun, 7=Sat): ");
        scanf("%d", &start_day);

        if (start_day < 1 || start_day > 7){
            printf("Invalid starting day of the week. Please input again.\n");
        }
    } while (start_day < 1 || start_day > 7);

    printf("\n");

    for (i = 1; i < start_day; i++){            // printing the blank days of the first week
        printf("   ");
    }

    for (j = 1; j <= days; i++, j++){           // printing the calendar numbers
        printf("%3d", j);
        if (i % 7 == 0){
            printf("\n");
        }
    }

    printf("\n\n");
    return 0;
}
```

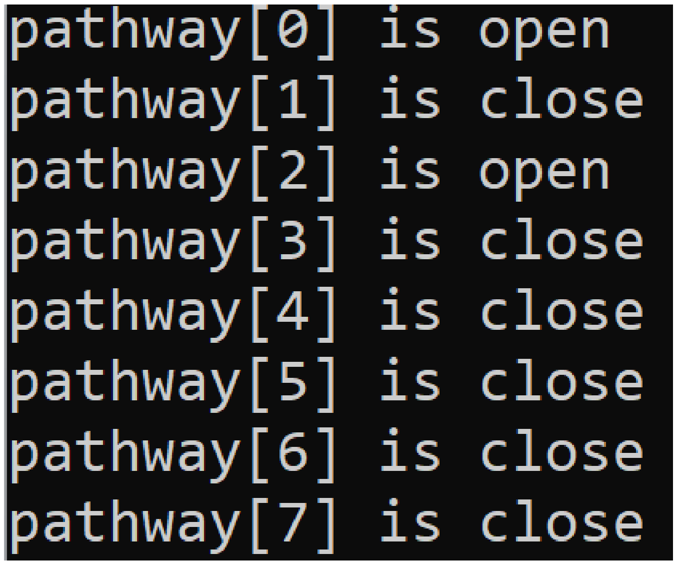


```
Enter number of days in the month: 31
Enter the starting day of the week (1=Sun, 7=Sat): 2

     1  2  3  4  5  6
  7  8  9 10 11 12 13
 14 15 16 17 18 19 20
 21 22 23 24 25 26 27
 28 29 30 31
```

6a)

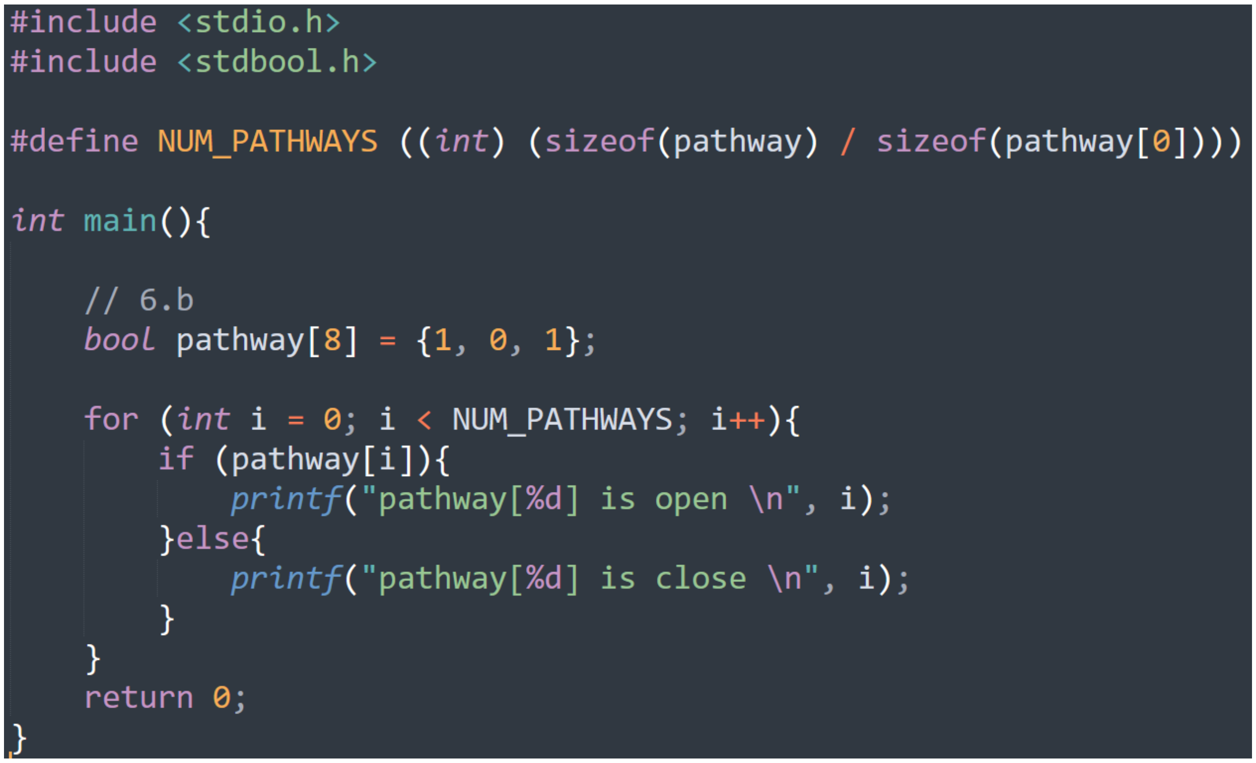


```c
#include <stdio.h>
#include <stdbool.h>

#define NUM_PATHWAYS ((int) (sizeof(pathway) / sizeof(pathway[0])))

int main(){

    // 6.a
    bool pathway[8] = {[0] = true, [2] = true};

    for (int i = 0; i < NUM_PATHWAYS; i++){
        if (pathway[i]){
            printf("pathway[%d] is open \n", i);
        }else{
            printf("pathway[%d] is close \n", i);
        }
    }

    return 0;
}
```

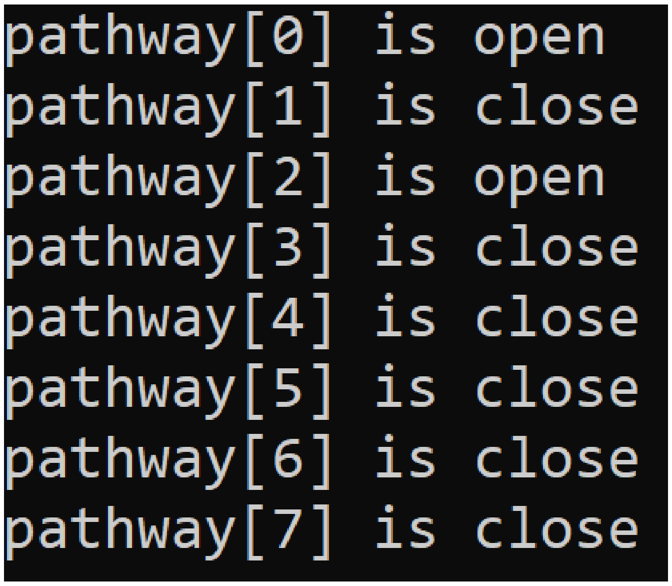


```
pathway[0] is open
pathway[1] is close
pathway[2] is open
pathway[3] is close
pathway[4] is close
pathway[5] is close
pathway[6] is close
pathway[7] is close
```

6b)

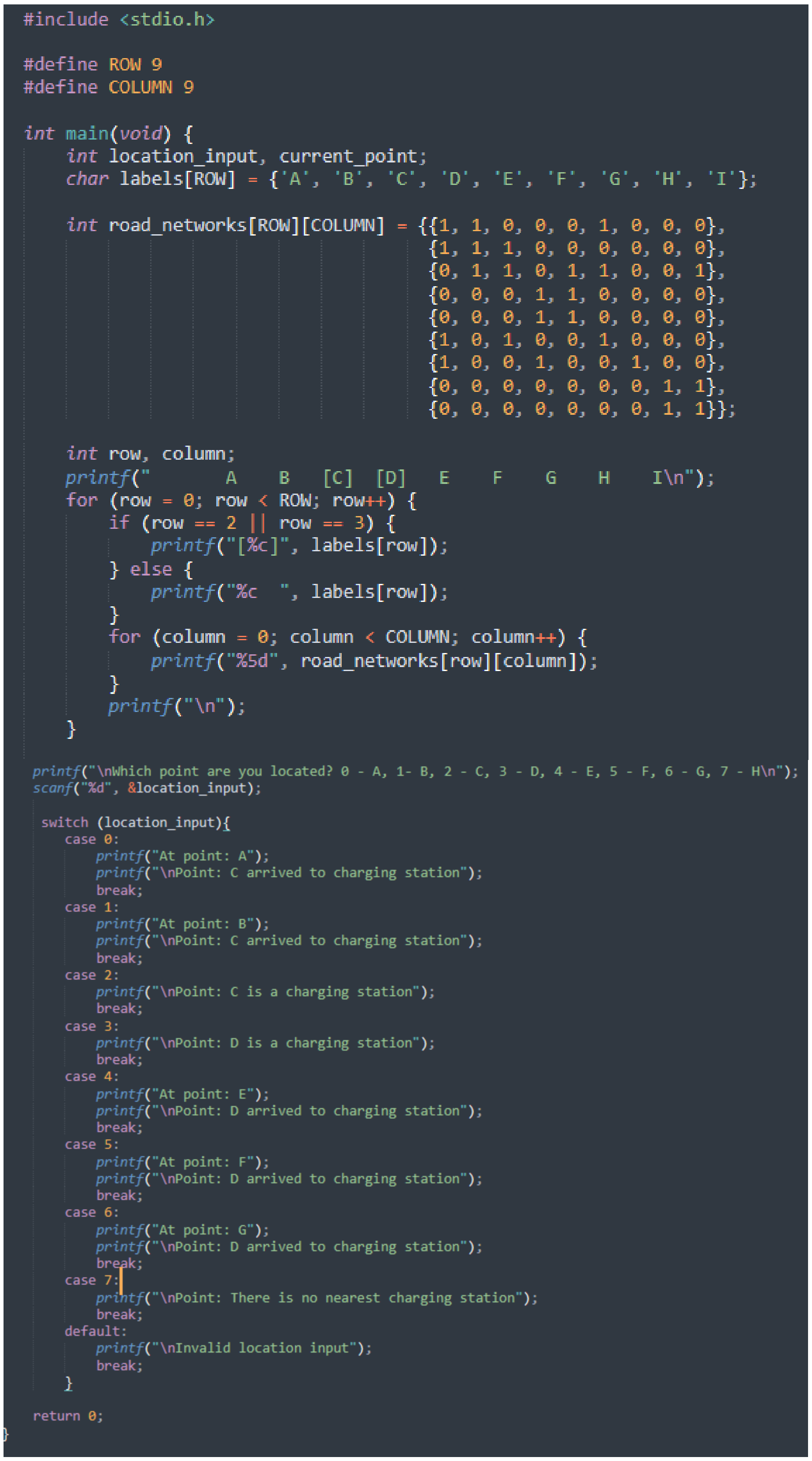


```c
#include <stdio.h>
#include <stdbool.h>

#define NUM_PATHWAYS ((int) (sizeof(pathway) / sizeof(pathway[0])))

int main(){

    // 6.b
    bool pathway[8] = {1, 0, 1};

    for (int i = 0; i < NUM_PATHWAYS; i++){
        if (pathway[i]){
            printf("pathway[%d] is open \n", i);
        }else{
            printf("pathway[%d] is close \n", i);
        }
    }
    return 0;
}
```

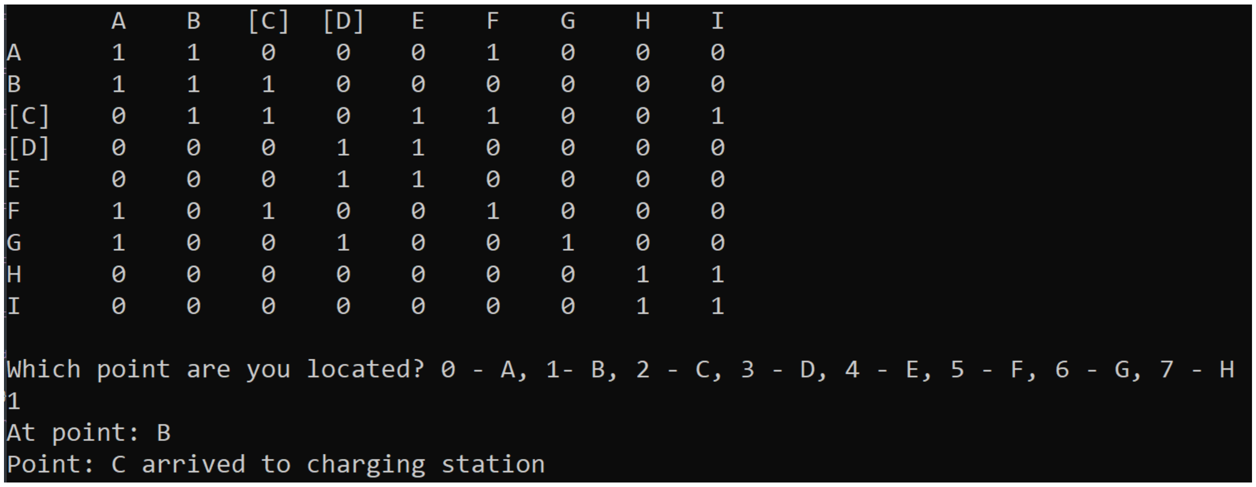


```
pathway[0] is open
pathway[1] is close
pathway[2] is open
pathway[3] is close
pathway[4] is close
pathway[5] is close
pathway[6] is close
pathway[7] is close
```

7)



```c
#include <stdio.h>

#define ROW 9
#define COLUMN 9

int main(void) {
    int location_input, current_point;
    char labels[ROW] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'};

    int road_networks[ROW][COLUMN] = {{1, 1, 0, 0, 0, 1, 0, 0, 0},
                                      {1, 1, 1, 0, 0, 0, 0, 0, 0},
                                      {0, 1, 1, 0, 1, 1, 0, 0, 1},
                                      {0, 0, 0, 1, 1, 0, 0, 0, 0},
                                      {0, 0, 0, 1, 1, 0, 0, 0, 0},
                                      {1, 0, 1, 0, 0, 1, 0, 0, 0},
                                      {1, 0, 0, 1, 0, 0, 1, 0, 0},
                                      {0, 0, 0, 0, 0, 0, 0, 1, 1},
                                      {0, 0, 0, 0, 0, 0, 0, 1, 1}};

    int row, column;
    printf("       A    B   [C]  [D]   E    F    G    H    I\n");
    for (row = 0; row < ROW; row++) {
        if (row == 2 || row == 3) {
            printf("[%c]", labels[row]);
        } else {
            printf("%c  ", labels[row]);
        }
        for (column = 0; column < COLUMN; column++) {
            printf("%5d", road_networks[row][column]);
        }
        printf("\n");
    }

 printf("\nWhich point are you located? 0 - A, 1- B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H\n");
 scanf("%d", &location_input);

  switch (location_input){
    case 0:
        printf("At point: A");
        printf("\nPoint: C arrived to charging station");
        break;
    case 1:
        printf("At point: B");
        printf("\nPoint: C arrived to charging station");
        break;
    case 2:
        printf("\nPoint: C is a charging station");
        break;
    case 3:
        printf("\nPoint: D is a charging station");
        break;
    case 4:
        printf("At point: E");
        printf("\nPoint: D arrived to charging station");
        break;
    case 5:
        printf("At point: F");
        printf("\nPoint: D arrived to charging station");
        break;
    case 6:
        printf("At point: G");
        printf("\nPoint: D arrived to charging station");
        break;
    case 7:
        printf("\nPoint: There is no nearest charging station");
        break;
    default:
        printf("\nInvalid location input");
        break;
    }

 return 0;
}
```



```
       A    B   [C]  [D]   E    F    G    H    I
A      1    1    0    0    0    1    0    0    0
B      1    1    1    0    0    0    0    0    0
[C]    0    1    1    0    1    1    0    0    1
[D]    0    0    0    1    1    0    0    0    0
E      0    0    0    1    1    0    0    0    0
F      1    0    1    0    0    1    0    0    0
G      1    0    0    1    0    0    1    0    0
H      0    0    0    0    0    0    0    1    1
I      0    0    0    0    0    0    0    1    1

Which point are you located? 0 - A, 1- B, 2 - C, 3 - D, 4 - E, 5 - F, 6 - G, 7 - H
1
At point: B
Point: C arrived to charging station
```